Self-adjusting entropy-stable scheme for compressible Euler equations*

Cheng Xiao-Han(程晓晗)[a], Nie Yu-Feng(聂玉峰)[a]†, Feng Jian-Hu(封建湖)[b], Luo Xiao-Yu[c], and Cai Li(蔡 力)[a]

[a] School of Science, Northwestern Polytechnical University, Xi'an 710129, China
[b] School of Science, Chang'an University, Xi'an 710064, China
[c] School of Mathematics and Statistics, University of Glasgow, Glasgow, G12 8QW, UK

(Received 17 July 2014; revised manuscript received 22 August 2014; published online 10 December 2014)

In this work, a self-adjusting entropy-stable scheme is proposed for solving compressible Euler equations. The entropy-stable scheme is constructed by combining the entropy conservative flux with a suitable diffusion operator. The entropy has to be preserved in smooth solutions and be dissipated at shocks. To achieve this, a switch function, which is based on entropy variables, is employed to make the numerical diffusion term be automatically added around discontinuities. The resulting scheme is still entropy-stable. A number of numerical experiments illustrating the robustness and accuracy of the scheme are presented. From these numerical results, we observe a remarkable gain in accuracy.

Keywords: compressible Euler equations, entropy-stable scheme, switch function

PACS: 02.30.Jr, 02.70.Bf **DOI:** 10.1088/1674-1056/24/2/020202

1. Introduction

In this paper, we focus on constructing an entropy-stable scheme for solving compressible Euler equations. These equations are of great importance since they govern the dynamics of a compressible material, such as gases and liquids at high pressures, while neglecting viscous and heat fluxes. Mathematically speaking, Euler equations can be written as a system of nonlinear hyperbolic conservation laws. As we all know, shock waves, or contact discontinuities, may be contained in solutions of Euler equations due to their nonlinear hyperbolic nature. Once discontinuities are present, the theories of classical solutions are no longer meaningful. Hence, solutions are interpreted in a weak sense. Although weak solutions allow for discontinuities, they may not be unique in general. Thus, an extra entropy condition must be imposed to select the physically relevant solution.[1] Numerical schemes which are consistent with the entropy condition are later termed entropy-stable. For Euler equations, the entropy condition is a direct representation of the second law of thermodynamics. That is to say, the entropy, which is conserved in smooth solutions, should be dissipated at shocks. A numerical method which does not satisfy the entropy condition may lead to nonphysical solutions, such as negative densities and pressures. For example, the famous Roe scheme[2] violates the entropy condition at sonic points. Therefore, various entropy fixes[3,4] have been proposed to modify the Roe scheme. These entropy fixes involve an artificial parameter that considerably depends on specific problems.

To achieve entropy stability, Tadmor introduced the concepts of entropy variable and entropy potential, and designed a class of entropy-conservative schemes, which are second-order accurate and conserve the entropy.[5,6] A criterion was also provided for whether a scheme is entropy-stable or not, namely: three-point schemes are entropy-stable if they contain more viscosity than entropy-conservative schemes. Following this criterion, one can devise entropy-stable schemes by adding suitable diffusion operators to entropy-conservative schemes. This idea was applied to Euler equations by using the dissipative fluxes of physical viscosity and heat conduction.[7] However, the result is satisfactory only on very fine meshes. Moreover, the computation of entropy-conservative flux for systems of conservation laws is complicated and numerically expensive. To overcome this obstacle, Ismail and Roe presented a simple and computationally efficient entropy-conservative flux for Euler equations.[8] A methodology for constructing high order (than second) entropy-stable schemes was developed by Fjordholm *et al.* by combining high order entropy-conservative fluxes and suitable numerical diffusion operators involving an ENO reconstruction which satisfies the sign property.[9–11]

In addition, other numerical schemes for Euler equations have been well-developed during the past few decades, such as the weighted essentially non-oscillatory (WENO) method,[12,13] the discontinuous Galerkin (DG) method,[14,15] the variational iteration method (VIM),[16–18] and the meshless method.[19,20] Among the variety of numerical meth-

*Project supported by the National Natural Science Foundation of China (Grant Nos. 11171043, 11101333, and 11471261) and the Doctorate Foundation of Northwestern Polytechnical University (Grant No. CX201426).

†Corresponding author. E-mail: yfnie@nwpu.edu.cn

© 2015 Chinese Physical Society and IOP Publishing Ltd

ods, WENO and DG methods have been studied and enjoyed widespread popularity. Developed from the successful ENO scheme,[21] WENO schemes use the nonlinear weights to automatically achieve high order accuracy in smooth regions of the solution and an essentially non-oscillatory resolution at the discontinuities. Many WENO-type methods[22–27] have been designed in recent years. The DG method was introduced by Reed and Hill[28] and further studied by Cockburn and Shu.[29–33] To advance in time with high order Runge–Kutta methods,[34] the resulting schemes were termed RKDG schemes. Different limiters[35–38] were constructed to suppress oscillations near shocks. However, to the best of our knowledge, rigorous stability results of these methods are still far from being complete, especially for nonlinear problems.

In the present article, we shall construct a self-adjusting entropy-stable scheme for compressible Euler equations. The new numerical flux contains two parts: entropy-conservative flux and a self-adjusting numerical diffusion term. The entropy-conservative part preserves the entropy exactly and performs well for smooth flows. However, it will give rise to oscillations in the vicinity of shocks. To eliminate such oscillations, a suitable numerical diffusion term is designed. Unlike the approach in Ref. [8], first order numerical diffusion is activated on the whole domain and makes the method less accurate in smooth regions. To avoid this, we utilize a switch function, which is based on entropy variables, to improve the accuracy of the entropy-stable method. The switch function's value possesses the property of being close to 1 at shocks and vanishing away from shocks so that the numerical diffusion works around discontinuities automatically. This is what we mean by 'self-adjusting'. It should be noted that the introduction of the switch function does not affect the entropy stability.

The rest of the paper is organized as follows. In Section 2 we give a brief overview of Euler equations and the related entropy-stable theory. Section 3 describes the detailed procedure for designing the self-adjusting entropy-stable scheme for solving compressible Euler equations. A variety of test problems are presented in Section 4, with the aim of assessing how well the scheme performs in practice. Our conclusions are drawn in Section 5.

2. Preliminaries

Without loss of generality, we consider the Euler equations for an ideal gas in two-dimensional (2D) space, written as a system of hyperbolic conservation laws,

$$U_t + F(U)_x + G(U)_y = 0, \tag{1}$$

where $U_t = \partial U/\partial t$, $F(U)_x = \partial F(U)/\partial x$, and $G(U)_y = \partial G(U)/\partial y$, and the U, F, G vectors given by

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \varepsilon \end{bmatrix}, \quad F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\varepsilon + p)u \end{bmatrix}, \quad G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\varepsilon + p)v \end{bmatrix}. \tag{2}$$

Here, ρ is the density, p is the pressure, and (u, v) are the velocity components in x and y directions. The total energy is defined as

$$\varepsilon = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2)$$

with the ideal gas constant $\gamma = 1.4$. This set of equations expresses the conservation laws for mass, momentum, and total energy. For simplicity, the spatial domain is discretized with a uniform mesh size $(\Delta x, \Delta y)$. We denote the grid points by (x_i, y_j) with $x_i = i\Delta x$ and $y_j = j\Delta y$. The domain is divided into cells $I_{i,j} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$ with midpoints $x_{i+1/2} = (x_i + x_{i+1})/2$ and $y_{j+1/2} = (y_j + y_{j+1})/2$. Then, a standard semi-discrete conservative finite difference (finite volume) scheme is given by

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} U_{i,j} = & -\frac{1}{\Delta x}\left(F_{i+1/2,j} - F_{i-1/2,j}\right) \\ & - \frac{1}{\Delta y}\left(G_{i,j+1/2} - G_{i,j-1/2}\right). \end{aligned} \tag{3}$$

Here, $U_{i,j}$ is the discrete solution at the grid point (x_i, y_j). $F_{i+1/2,j}$ and $G_{i,j+1/2}$ are numerical fluxes consistent with F and G, respectively.

Assume that system (1) is equipped with a convex entropy function $E = E(U)$, and associated entropy flux functions $P = P(U)$ and $Q = Q(U)$, such that the following compatibility conditions hold

$$P'(U)^{\mathrm{T}} = E'(U)^{\mathrm{T}} F'(U), \quad Q'(U)^{\mathrm{T}} = E'(U)^{\mathrm{T}} G'(U). \tag{4}$$

For brevity of notions $(\bullet)'$ is written for $\partial(\bullet)/\partial U$ here. Then, it is straightforward to check that smooth solutions of Eq. (1) satisfy an additional conservation laws, also known as the entropy equality,

$$E(U)_t + P(U)_x + Q(U)_y = 0. \tag{5}$$

But for discontinuous solutions, the entropy has to be dissipated by implying the entropy inequality

$$E(U)_t + P(U)_x + Q(U)_y \leq 0 \tag{6}$$

(in a weak sense). Next, the formula (6) will be used as an essential component in our strategy to yield stable numerical schemes for Eq. (1) in this paper.

3. Self-adjusting entropy-stable scheme

3.1. Entropy-conservative flux

In this subsection, we describe the entropy-conservative scheme for 2D Euler equations. Here, entropy conservation

means that the computed solutions satisfy the discrete entropy equality

$$\frac{\mathrm{d}}{\mathrm{d}t}E(U_{i,j})+\frac{1}{\Delta x}(\tilde{P}_{i+1/2,j}-\tilde{P}_{i-1/2,j})$$
$$\frac{1}{\Delta y}(\tilde{Q}_{i,j+1/2}-\tilde{Q}_{i,j-1/2})=0 \tag{7}$$

with numerical entropy fluxes $\tilde{P}_{i+1/2,j}$ and $\tilde{Q}_{i,j+1/2}$ consistent with P and Q. For convenience, let us introduce the following notation

$$\begin{aligned}
[W]_{i+1/2,j}&=W_{i+1,j}-W_{i,j},\\
\overline{W}_{i+1/2,j}&=\frac{1}{2}(W_{i+1,j}+W_{i,j}),\\
[W]_{i,j+1/2}&=W_{i,j+1}-W_{i,j},\\
\overline{W}_{i,j+1/2}&=\frac{1}{2}(W_{i,j+1}+W_{i,j}).
\end{aligned} \tag{8}$$

Define the entropy variables as

$$V(U)=\frac{\partial E}{\partial U} \tag{9}$$

and the entropy potentials as

$$\begin{aligned}
\Psi(U)&=V(U)^{\mathrm{T}}F(U)-P(U),\\
\Phi(U)&=V(U)^{\mathrm{T}}G(U)-Q(U).
\end{aligned} \tag{10}$$

Then, analogous to the one-dimensional (1D) case in Ref. [6], we give the two-dimensional entropy-conservative scheme by stating the following.

Proposition 1 The scheme (3) is entropy-conservative if the numerical fluxes $\tilde{F}_{i+1/2,j}$ and $\tilde{G}_{i,j+1/2}$ satisfy the conditions

$$\begin{aligned}
[V]^{\mathrm{T}}_{i+1/2,j}\tilde{F}_{i+1/2,j}&=[\Psi]_{i+1/2,j},\\
[V]^{\mathrm{T}}_{i,j+1/2}\tilde{G}_{i,j+1/2}&=[\Phi]_{i,j+1/2}
\end{aligned} \tag{11}$$

with numerical entropy fluxes

$$\begin{aligned}
\tilde{P}_{i+1/2,j}&=\overline{V}^{\mathrm{T}}_{i+1/2,j}F_{i+1/2,j}-\overline{\Psi}_{i+1/2,j},\\
\tilde{Q}_{i,j+1/2}&=\overline{V}^{\mathrm{T}}_{i,j+1/2}G_{i,j+1/2}-\overline{\Phi}_{i,j+1/2}.
\end{aligned} \tag{12}$$

For Euler equations, the entropy pair is chosen to be

$$E=-\frac{\rho s}{\gamma-1},\quad P=-\frac{\rho us}{\gamma-1},\quad Q=-\frac{\rho vs}{\gamma-1}, \tag{13}$$

where s is the physical entropy given by $s=\ln p-\gamma\ln\rho$. The above-defined entropy pair gives a simple form of entropy variables and entropy potentials

$$\begin{aligned}
V&=\begin{bmatrix}(\gamma-s)/(\gamma-1)-\rho(u^2+v^2)/2p\\ \rho u/p\\ \rho v/p\\ -\rho/p\end{bmatrix},\\
\Psi(U)&=\rho u,\quad \Phi(U)=\rho v.
\end{aligned} \tag{14}$$

In general, the relations (11) provide two algebraic equations for several unknowns. Hence, the solutions of Eq. (11) are not unique. Although there are mainly three ways to compute entropy-conservative numerical fluxes for Euler equations, we will only describe Roe's explicit solutions here. Defining the parameter vectors $\boldsymbol{z}$ as

$$\boldsymbol{z}=\begin{bmatrix}z^1\\ z^2\\ z^3\\ z^4\end{bmatrix}=\sqrt{\frac{\rho}{p}}\begin{bmatrix}1\\ u\\ v\\ p\end{bmatrix}, \tag{15}$$

then the entropy-conservative fluxes in x and y directions are expressed by $\tilde{F}_{i+1/2,j}=[\tilde{F}^1_{i+1/2,j},\tilde{F}^2_{i+1/2,j},\tilde{F}^3_{i+1/2,j},\tilde{F}^4_{i+1/2,j}]^{\mathrm{T}}$ and $\tilde{G}_{i,j+1/2}=[\tilde{G}^1_{i,j+1/2},\tilde{G}^2_{i,j+1/2},\tilde{G}^3_{i,j+1/2},\tilde{G}^4_{i,j+1/2}]^{\mathrm{T}}$ with

$$\begin{aligned}
\tilde{F}^1_{i+1/2,j}&=\overline{z^2}_{i+1/2,j}(z^4)^{\ln}_{i+1/2,j},\\
\tilde{F}^2_{i+1/2,j}&=\frac{\overline{z^4}_{i+1/2,j}}{\overline{z^1}_{i+1/2,j}}+\frac{\overline{z^2}_{i+1/2,j}}{\overline{z^1}_{i+1/2,j}}\tilde{F}^1_{i+1/2,j},\\
\tilde{F}^3_{i+1/2,j}&=\frac{\overline{z^3}_{i+1/2,j}}{\overline{z^1}_{i+1/2,j}}\tilde{F}^1_{i+1/2,j},\\
\tilde{F}^4_{i+1/2,j}&=\frac{1}{2\overline{z^1}_{i+1/2,j}}\left(\frac{\gamma+1}{\gamma-1}\frac{\tilde{F}^1_{i+1/2,j}}{(z^1)^{\ln}_{i+1/2,j}}+\overline{z^2}_{i+1/2,j}\tilde{F}^2_{i+1/2,j}\right.\\
&\quad\left.+\overline{z^3}_{i+1/2,j}\tilde{F}^3_{i+1/2,j}\right),
\end{aligned} \tag{16}$$

and

$$\begin{aligned}
\tilde{G}^1_{i,j+1/2}&=\overline{z^3}_{i,j+1/2}(z^4)^{\ln}_{i,j+1/2},\\
\tilde{G}^2_{i,j+1/2}&=\frac{\overline{z^2}_{i,j+1/2}}{\overline{z^1}_{i,j+1/2}}\tilde{G}^1_{i,j+1/2},\\
\tilde{G}^3_{i,j+1/2}&=\frac{\overline{z^4}_{i,j+1/2}}{\overline{z^1}_{i,j+1/2}}+\frac{\overline{z^3}_{i,j+1/2}}{\overline{z^1}_{i,j+1/2}}\tilde{G}^1_{i,j+1/2},\\
\tilde{G}^4_{i,j+1/2}&=\frac{1}{2\overline{z^1}_{i,j+1/2}}\left(\frac{\gamma+1}{\gamma-1}\frac{\tilde{G}^1_{i,j+1/2}}{(z^1)^{\ln}_{i,j+1/2}}+\overline{z^2}_{i,j+1/2}\tilde{G}^2_{i,j+1/2}\right.\\
&\quad\left.+\overline{z^3}_{i,j+1/2}\tilde{G}^3_{i,j+1/2}\right).
\end{aligned} \tag{17}$$

Here, $a^{\ln}_{i+1/2,j}$ and $a^{\ln}_{i,j+1/2}$ denote the logarithmic means defined as

$$a^{\ln}_{i+1/2,j}=\frac{[a]_{i+1/2,j}}{[\ln(a)]_{i+1/2,j}},\quad a^{\ln}_{i,j+1/2}=\frac{[a]_{i,j+1/2}}{[\ln(a)]_{i,j+1/2}}. \tag{18}$$

An efficient computation of the logarithmic means is also given in Ref. [8]. We would like to mention that Roe's entropy-conservative numerical flux is not numerically expensive and is well-formed.

3.2. Self-adjusting numerical diffusion operator

Entropy-conservative scheme preserves the entropy and performs well in smooth regions. However, it will produce oscillations near shocks. Thus, numerical diffusion operators should be added to ensure entropy stability. A scheme is said to be entropy-stable if the computed solutions satisfy the discrete entropy inequality

$$\frac{\mathrm{d}}{\mathrm{d}t}E(U_{i,j})+\frac{1}{\Delta x}(P_{i+1/2,j}-P_{i-1/2,j}) + \frac{1}{\Delta y}(Q_{i,j+1/2}-Q_{i,j-1/2})\leq 0. \tag{19}$$

Define the following numerical fluxes,

$$F_{i+1/2,j}=\tilde{F}_{i+1/2,j}-\frac{1}{2}D^{x}_{i+1/2,j}[V]_{i+1/2,j},$$
$$G_{i,j+1/2}=\tilde{G}_{i,j+1/2}-\frac{1}{2}D^{y}_{i,j+1/2}[V]_{i,j+1/2}, \tag{20}$$

where $D^{x}_{i+1/2,j}$ and $D^{y}_{i,j+1/2}$ are symmetric positive-definite matrices.

Proposition 2 The scheme (3) with numerical fluxes (20) is entropy-stable with the numerical entropy fluxes given by

$$P_{i+1/2,j}=\tilde{P}_{i+1/2,j}-\frac{1}{2}\overline{V}^{\mathrm{T}}_{i+1/2,j}D^{x}_{i+1/2,j}[V]_{i+1/2,j},$$
$$Q_{i,j+1/2}=\tilde{Q}_{i,j+1/2}-\frac{1}{2}\overline{V}^{\mathrm{T}}_{i,j+1/2}D^{y}_{i,j+1/2}[V]_{i,j+1/2}.$$

Next, we will discuss the diffusion matrices in more detail. For simplicity, we will focus on the diffusion matrix $D^{x}_{i+1/2,j}$ (in x direction) only. A similar procedure can be carried out easily in y direction. We choose $D^{x}_{i+1/2,j}$ to be the following form

$$D^{x}_{i+1/2,j}=R_{i+1/2,j}\Theta_{i+1/2,j}\Lambda_{i+1/2,j}S_{i+1/2,j}R^{\mathrm{T}}_{i+1/2,j}. \tag{21}$$

Here, $R_{i+1/2,j}$ is the averaged state of the right eigenvectors of $\partial F/\partial U$,

$$R_{i+1/2,j}=\begin{bmatrix}1 & 0 & 1 & 1\\ u-c & 0 & u & u+c\\ v & 1 & v & v\\ H-uc & v & (u^2+v^2)/2 & H+uc\end{bmatrix}. \tag{22}$$

$\Lambda_{i+1/2,j}$ is the diagonal matrix of absolute eigenvalues,

$$\Lambda_{i+1/2,j}=\mathrm{diag}\left(|u-c|,|u|,|u|,|u+c|\right). \tag{23}$$

$S_{i+1/2,j}$ is a matrix of scaling factors,

$$S_{i+1/2,j}=\mathrm{diag}\left(\frac{\rho}{2\gamma},\frac{(\gamma-1)\rho}{\gamma},p,\frac{\rho}{2\gamma}\right) \tag{24}$$

such that $R^{-1}\mathrm{d}U=SR^{\mathrm{T}}\mathrm{d}V$. All the averaged values in $R_{i+1/2,j}$, $\Lambda_{i+1/2,j}$, and $S_{i+1/2,j}$ are determined exactly as in Ref. [8]. The remained $\Theta_{i+1/2,j}$, which is defined as

$$\Theta_{i+1/2,j}=\mathrm{diag}(\theta^{1}_{i+1/2,j},\theta^{2}_{i+1/2,j},\theta^{3}_{i+1/2,j},\theta^{4}_{i+1/2,j}), \tag{25}$$

is a switch matrix that signals the amount of numerical diffusion to be added to the entropy-conservative scheme corresponding to each wave. We set $\theta^{l}_{i+1/2,j}\,(l=1,2,3,4)$ to be a constant, $0\leqslant\theta^{l}_{i+1/2,j}\leqslant 1$. More specifically, $\theta^{l}_{i+1/2,j}\approx 1$ near shocks. But away from the shocks it can be very small, so that the effect can be neglected. By using this mechanism the entropy-conservative scheme is activated at all times and numerical diffusion is added automatically near shocks. In this sense, the scheme is 'self-adjusting'. We set

$$\theta^{l}_{i+1/2,j}=\max(\hat{\theta}^{l}_{i,j},\hat{\theta}^{l}_{i+1,j}), \tag{26}$$

and use the switch function proposed by Jamson[39]

$$\hat{\theta}^{l}_{i,j}=\left|\frac{\alpha^{l}_{i+1/2,j}-\alpha^{l}_{i-1/2,j}}{\left|\alpha^{l}_{i+1/2,j}\right|+\left|\alpha^{l}_{i-1/2,j}\right|}\right|^{p}. \tag{27}$$

The $\alpha^{l}_{i+1/2,j}$ are elements of $R^{\mathrm{T}}_{i+1/2,j}(V_{i+1,j}-V_{i,j})$. By varying the $p\geqslant 1$ in Eq. (27), one can change the order of accuracy of the diffusion term. It is easy to see $\hat{\theta}^{l}_{i,j}=O((\Delta x)^{p})$ wherever the solution is smooth, so our scheme is at least second-order in smooth regions. (We take p of 2 in our numerical experiments.)

Remark 1 If one takes $D^{x}_{i+1/2,j}=R_{i+1/2,j}\Lambda_{i+1/2,j}S_{i+1/2,j}R^{\mathrm{T}}_{i+1/2,j}$, then the scheme reduces to the same as in Ref. [8]. Since the jump term $[V]$ is of the order $O(\Delta x)$, the scheme's accuracy is lowered in smooth regions. Although a second-order TVD version is used to improve the accuracy, the resulting scheme may not be entropy-stable. In our scheme, the introduction of a switch matrix $\Theta_{i+1/2,j}$ does not affect the entropy stability while maintaining at least second-order in smooth regions.

Remark 2 To generate 'enough' entropy production across a shock, Ismail and Roe proposed to modify the eigenvalue matrix $\Lambda_{i+1/2,j}$ and the resulting fluxes were called entropy-consistent fluxes.[8]

4. Numerical examples

In the following, a number of typical numerical examples are provided to illustrate the robustness and accuracy of the proposed scheme (ERoe-J). The results computed by the entropy-stable scheme (ERoe)[8] are also presented for a comparison. To complete the computation, a proper time discretization is needed. In the current work, we advance in time with the strong stability-preserving third-order Runge–Kutta (RK3) method,[40]

$$\begin{cases}U^{(1)}=U^{n}+\Delta tL(U^{n}),\\ U^{(2)}=\frac{3}{4}U^{n}+\frac{1}{4}U^{(1)}+\frac{1}{4}\Delta tL(U^{(1)}),\\ U^{n+1}=\frac{1}{3}U^{n}+\frac{2}{3}U^{(2)}+\frac{2}{3}\Delta tL(U^{(2)}),\end{cases} \tag{28}$$

where

$$[L(U)]_{i,j} := -\frac{1}{\Delta x}\left(F_{i+1/2,j}-F_{i-1/2,j}\right) - \frac{1}{\Delta y}\left(G_{i,j+1/2}-G_{i,j-1/2}\right). \quad (29)$$

In all of the experiments below, the CFL number is taken as 0.3 and the time step is evaluated by

$$\Delta t = \frac{CFL}{\max_{i,j}((c_{i,j}+|u_{i,j}|)/\Delta x,(c_{i,j}+|v_{i,j}|)/\Delta y)}$$

with the local sound speed $c_{i,j}=\sqrt{\gamma p_{i,j}/\rho_{i,j}}$.

4.1. 1D problems

Example 1 To test the accuracy of the proposed scheme, we consider the 1D Euler equations

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho u\\ \varepsilon\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\rho u\\ \rho u^2+p\\ (\varepsilon+p)u\end{bmatrix}=0 \quad (30)$$

with the initial data

$$\rho = 1+0.2\sin(\pi x),\ u=1,\ p=1$$

and a 2-period boundary conditions. The exact solution is $\rho(x,t)=1+0.2\sin(\pi(x-t))$, $u(x,t)=1$, $p(x,t)=1$. We compute the solution up to time $t=2$. Table 1 presents the errors and convergence rates computed by ERoe and ERoe-J, respectively. It can be seen that ERoe-J gives errors of a smaller magnitude when compared with ERoe and achieves second-order accuracy in L^1 norm. Thanks to the switch function the first-order numerical diffusion does not affect the overall accuracy of the scheme.

Table 1. The errors and convergence rates of density ρ at $t=2$.

Method	N	L^1 error	L^1 order	L^∞ error	L^∞ order
ERoe	20	1.604×10^{-1}	–	1.253×10^{-1}	–
	40	9.935×10^{-2}	0.691	7.783×10^{-2}	0.687
	80	5.571×10^{-2}	0.835	4.372×10^{-2}	0.832
	160	2.956×10^{-2}	0.914	2.321×10^{-2}	0.914
	320	1.523×10^{-2}	0.957	1.196×10^{-2}	0.899
	640	7.734×10^{-3}	0.978	6.074×10^{-3}	0.977
ERoe-J	20	4.479×10^{-2}	–	3.977×10^{-2}	–
	40	1.428×10^{-2}	1.649	1.559×10^{-2}	1.351
	80	4.069×10^{-3}	1.811	5.862×10^{-3}	1.411
	160	1.067×10^{-3}	1.931	2.168×10^{-3}	1.435
	320	2.703×10^{-4}	1.981	8.241×10^{-4}	1.395
	640	6.754×10^{-5}	2.001	3.245×10^{-4}	1.345

Example 2 Consider Sod's shock tube problem[41] with the initial data:

$$(\rho,u,p)=\begin{cases}(1,0,1), & \text{for } x<0,\\ (0.125,0,0.1), & \text{for } x>0.\end{cases}$$

This initial continuity evolves into a left-going rarefaction wave, a right-going shock wave, and a right-going contact discontinuity. We compute the solution at time $t=0.16$ on a mesh of 100 points. The obtained density is shown in Fig. 1. As one can clearly see, ERoe-J resolves all the waves better than ERoe.

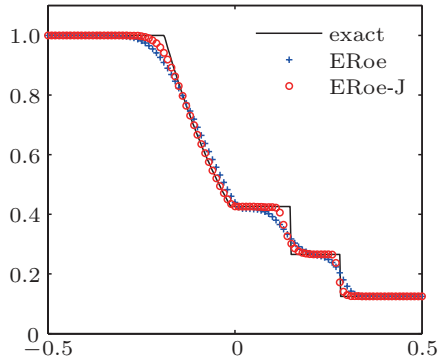


Fig. 1. (color online) Density of Sod's shock tube problem.

Example 3 Consider Toro's 123 problem[42] subject to the initial data:

$$(\rho,u,p)=\begin{cases}(1,-2,0.4), & \text{for } x<0,\\ (1,2,0.4), & \text{for } x>0.\end{cases}$$

The exact solution consists of two strong rarefaction waves and a trivial stationary contact discontinuity. Between the rarefactions, the pressure is very small (near vacuum) and this can lead to difficulties in numerical simulations. We compute the solution at time $t=0.1$ on a mesh of 400 points. Figure 2 displays the results of density and internal energy. One can easily observe that ERoe-J produces a more accurate solution.

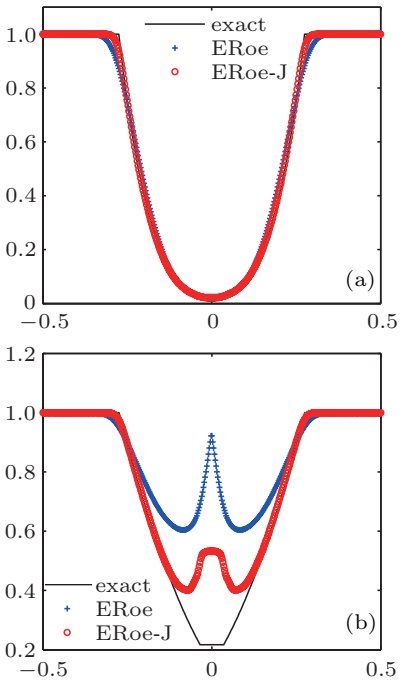


Fig. 2. (color online) Solutions of Toro's 123 problem: (a) density and (b) internal energy.

4.2. 2D problems

Example 4 We solve a long-time vortex advection problem[43] which describes the perturbations of a mean flow. The initial data are given by

$$u = 1-(y-y_c)\phi(r),$$
$$v = 1+(x-x_c)\phi(r),$$
$$T = 1-\frac{\gamma-1}{2\gamma}\phi(r)^2, s = 0$$

with

$$\phi(r) = \frac{5}{2\pi}e^{0.5(1-r^2)}, \quad r = \sqrt{(x-x_c)^2+(y-y_c)^2}.$$

Here, the temperature is defined as $T = p/\rho$ and s is the physical entropy that was mentioned before. The initial center of the vortex is taken as $(x_c, y_c) = (5,5)$. The exact solution to this initial value problem is just the passive convection of the vortex with a velocity of 1 in x and y directions, i.e.

$$U(x,y,t) = U(x-t, y-t, 0).$$

The computational domain is chosen to be $[0,10]\times[0,10]$ and we extend periodically in both directions. It is shown in Table 2 that ERoe-J achieves second-order accuracy in the 2D case. The computation is also performed on a mesh of 200×200 grid up to time $t = 100$. As one can see from Fig. 3, ERoe does not give a correct solution after long time simulation. To the contrary, ERoe-J preserves the vortex well.

Table 2. The errors and convergence rates of density ρ at $t = 10$.

Method	Grids	L^1 error	L^1 order	L^∞ error	L^∞ order
ERoe-J	60× 60	4.664×10^{-1}	–	1.174×10^{-1}	–
	120× 120	1.154×10^{-1}	2.016	3.128×10^{-2}	1.908
	240× 240	2.619×10^{-2}	2.139	7.206×10^{-3}	2.118

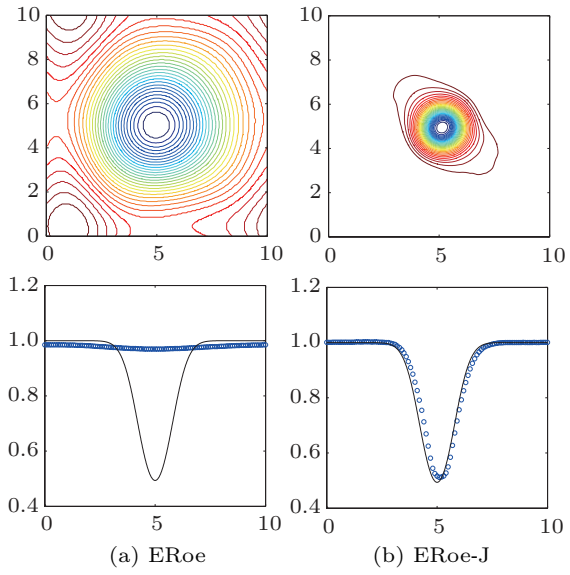


Fig. 3. (color online) Solutions of the long-time vortex advection problem. Top: Plots of 30 density contour lines. Bottom: A density slice in x direction at $y = 5$ (numerical solution: circles, exact solution: line).

Example 5 Consider the Riemann problems with initial data on $[0,1]\times[0,1]$ of the form

$$(p,\rho,u,v)(x,y,0) = \begin{cases} (p_1,\rho_1,u_1,v_1), & \text{if } x>0.5, y>0.5,\\ (p_2,\rho_2,u_2,v_2), & \text{if } x<0.5, y>0.5,\\ (p_3,\rho_3,u_3,v_3), & \text{if } x<0.5, y<0.5,\\ (p_4,\rho_4,u_4,v_4), & \text{if } x>0.5, y<0.5. \end{cases}$$

According to Ref. [44], there are 19 different admissible configurations separated by rarefaction (R), shock (S), and contact wave (J). We only show the results for the following two configurations.

Configuration 4 (Four shock waves)

$$p_2 = 0.35, \quad \rho_2 = 0.5065, \quad p_1 = 1.1, \quad \rho_1 = 1.1,$$
$$u_2 = 0.8939, \quad v_2 = 0, \quad u_1 = 0, \quad v_1 = 0,$$
$$p_3 = 1.1, \quad \rho_3 = 1.1, \quad p_4 = 0.35, \quad \rho_4 = 0.5065,$$
$$u_3 = 0.8939, \quad v_3 = 0.8939, \quad u_4 = 0, \quad v_4 = 0.8939.$$

Configuration 19 (One shock wave, one rarefaction wave, and two contacts):

$$p_2 = 1, \quad \rho_2 = 2, \quad p_1 = 1, \quad \rho_1 = 1,$$
$$u_2 = 0, \quad v_2 = -0.3, \quad u_1 = 0, \quad v_1 = 0.3,$$
$$p_3 = 0.4, \quad \rho_3 = 1.0625, \quad p_4 = 0.4, \quad \rho_4 = 0.5197,$$
$$u_3 = 0 \quad v_3 = 0.2145, \quad u_4 = 0, \quad v_4 = -0.4259.$$

The computations are done on a 400×400 grid and the density contour lines are shown in Figs. 4 and 5, respectively. Taking the results in Ref. [44] as reference solutions, we notice that ERoe-J achieves much better resolution than ERoe. For configuration 4, ERoe-J captures the shocks with less intermediate cells. And for configuration 19, ERoe-J resolves the wave structure more accurately (see the zoomed regions).

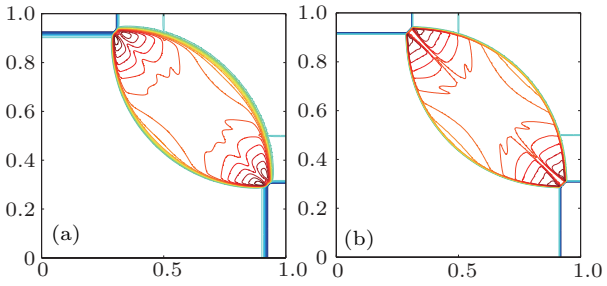


Fig. 4. (color online) Configuration 4 with 30 density contour lines, $t = 0.25$: (a) ERoe, (b) ERoe-J.

Example 6 We solve the double Mach reflection problem, originally from Ref. [45]. This is a standard test case for high resolution schemes. The computational domain is set to be [0,4]×[0,1] and the final time is $t = 0.2$. The reflecting wall lies along the bottom of the domain, beginning at $x = 1/6$. Initially, a right-moving Mach 10 shock, positioned at $(x = 1/6, y = 0)$, makes a 60° angle with the reflecting wall and extends to the top of the domain. For the bottom boundary, the post-shock condition is imposed from $x = 0$ to $x = 1/6$ and

a reflective condition is set from $x = 1/6$ to $x = 4$. At the top boundary, the values are set to describe the exact motion of a Mach-10 shock. The left and right boundaries are assigned to the inflow and outflow conditions, respectively. We consider a 480×120 grid for this problem. The density contours of the solution are shown in Fig. 6. The WENO-LF-5 results[46] are provided as a reference solution. It is observed that ERoe produces a kinked Mach stem but ERoe-J avoids such a failure.

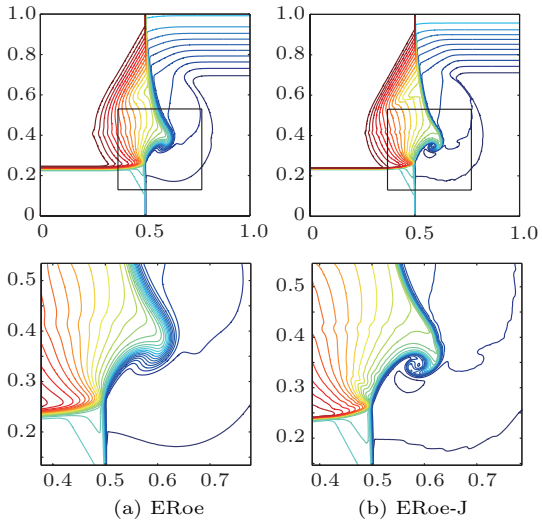


Fig. 5. (color online) Solutions of configuration 19 at $t = 0.3$. Top: Plots of 30 density contour lines. Bottom: Zoomed regions.

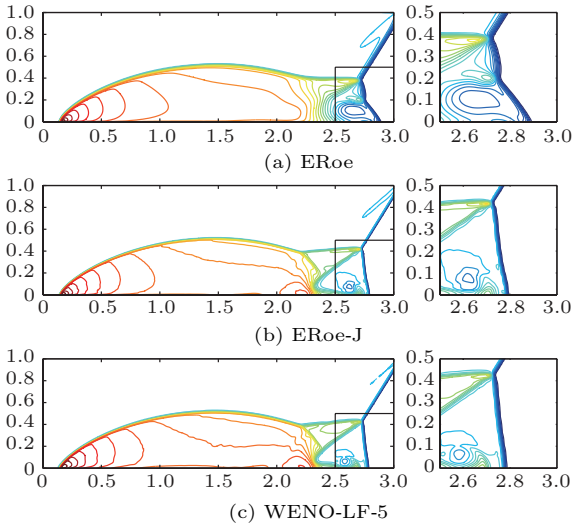


Fig. 6. (color online) Solutions of double Mach reflection problem. Left: Plots of 30 density contour lines. Right: Zoomed regions.

5. Conclusions

In this paper, we have proposed a self-adjusting entropy-stable scheme to solve compressible Euler equations. Following Tadmor's idea, an entropy-stable scheme is designed by adding a suitable numerical diffusion to the entropy-conservative flux. To achieve self-adjustment, we introduce a switch matrix which signals the amount of numerical diffusion so that numerical diffusion is added to the entropy-conservative flux automatically. Our numerical experiments demonstrate a remarkable gain in accuracy of the proposed scheme.

References

[1] Dafermos C M 2000 *Conservation Laws in Continuum Physics* (Berlin: Springer)
[2] Roe P L 1981 *J. Comput. Phys.* **43** 357
[3] Harten A 1983 *J. Comput. Phys.* **49** 357
[4] Yee H C, Klopfer G H and Montagne J L 1990 *J. Comput. Phys.* **88** 31
[5] Tadmor E 1987 *Math. Comput.* **49** 91
[6] Tadmor E 2003 *Acta Numerica* **12** 451
[7] Tadmor E and Zhong W G 2006 *J. Hyperbolic Diff. Eq.* **3** 529
[8] Ismail F and Roe P L 2009 *J. Comput. Phys.* **228** 5410
[9] Lefloch P G, Mercier J M and Rohde C 2002 *SIAM J. Numer. Anal.* **40** 1968
[10] Fjordholm U S, Mishra S and Tadmor E 2012 *SIAM J. Numer. Anal.* **50** 544
[11] Fjordholm U S, Mishra S and Tadmor E 2013 *Found. Comput. Math.* **13** 139
[12] Liu X D, Osher S and Chan T 1994 *J. Comput. Phys.* **115** 200
[13] Shu C W 2009 *SIAM Rev.* **51** 82
[14] Zhao G Z, Yu X J and Guo P Y 2013 *Chin. Phys. B* **22** 020202
[15] Zhao G Z, Yu X J and Zhang R P 2013 *Chin. Phys. B* **22** 050206
[16] He J H 1999 *Int. J. Nonlinear Mech.* **34** 699
[17] Mo J Q and Lin W T 2008 *Acta Phys. Sin.* **57** 1291 (in Chinese)
[18] Zhao G Z, Yu X J, Xu Y and Zhu J 2010 *Chin. Phys. B* **19** 070203
[19] Cheng R J and Cheng Y M 2011 *Chin. Phys. B* **20** 070206
[20] Ge H X, Liu Y Q and Cheng R J 2012 *Chin. Phys. B* **21** 010206
[21] Harten A, Engquist B, Osher S and Chakravarthy S R 1987 *J. Comput. Phys.* **71** 231
[22] Borges R, Carmona M, Costa B and Don W S 2008 *J. Comput. Phys.* **227** 3191
[23] Hu X Y, Wang Q and Adams N A 2010 *J. Comput. Phys.* **229** 8952
[24] Castro M, Costa B and Don W S 2011 *J. Comput. Phys.* **230** 1766
[25] Zhang X X and Shu C W 2012 *J. Comput. Phys.* **231** 2245
[26] Fan P 2014 *J. Comput. Phys.* **269** 355
[27] Huang C S, Arbogast T and Hung C H 2014 *J. Comput. Phys.* **262** 291
[28] Reed W H and Hill T R 1973 *Triangular Mesh Methods for the Neutron Transport Equation* (Los Alamos Scientific Laboratory Report LA-UR-73-479)
[29] Cockburn B and Shu C W 1991 *Math. Model. Numer. Anal.* (M2AN) **25** 337
[30] Cockburn B and Shu C W 1989 *Math. Comp.* **52** 411
[31] Cockburn B, Lin S Y and Shu C W 1989 *J. Comput. Phys.* **84** 90
[32] Cockburn B, Hou S and Shu C W 1990 *Math. Comp.* **54** 545
[33] Cockburn B and Shu C W 1998 *J. Comput. Phys.* **141** 199
[34] Gottlieb S, Ketcheson D and Shu C W 2011 *Strong Stability Preserving Runge–Kutta and Multistep Time Discretizations* (Singapore: World Scientific)
[35] Qiu J X and Shu C W 2005 *SIAM J. Sci. Comput.* **26** 907
[36] Zhong X and Shu C W 2013 *J. Comput. Phys.* **232** 397
[37] Zhu J and Qiu J X 2013 *SIAM J. Sci. Comput.* **55** 606
[38] Zhu J, Zhong X, Shu C W and Qiu J X 2013 *J. Comput. Phys.* **248** 200
[39] Jameson A 2008 *J. Sci. Comput.* **34** 152
[40] Gottlieb S, Shu C W and Tadmor E 2001 *SIAM Rev.* **43** 89
[41] Sod G A 1978 *J. Comput. Phys.* **27** 1
[42] Toro E F 2009 *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction* (Berlin: Springer)
[43] Shu C W 1998 *Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws* (Berlin: Springer)
[44] Lax P D and Liu X D 1998 *SIAM J. Sci. Comput.* **19** 319
[45] Woodward P and Colella P 1984 *J. Comput. Phys.* **54** 115
[46] Jiang G S and Shu C W 1996 *J. Comput. Phys.* **126** 202